

Parameterized approximation algorithms for bidirected steiner network problems

Chitnis, Rajesh; Feldmann, Andreas Emil; Manurangsi, Pasin

DOI:

[10.4230/LIPIcs.ESA.2018.20](https://doi.org/10.4230/LIPIcs.ESA.2018.20)

License:

Creative Commons: Attribution (CC BY)

Document Version

Publisher's PDF, also known as Version of record

Citation for published version (Harvard):

Chitnis, R, Feldmann, AE & Manurangsi, P 2018, Parameterized approximation algorithms for bidirected steiner network problems. in H Bast, G Herman & Y Azar (eds), *26th European Symposium on Algorithms, (ESA 2018)*., 20, Leibniz International Proceedings in Informatics, LIPIcs, vol. 112, Schloss Dagstuhl, 26th European Symposium on Algorithms, ESA 2018, Helsinki, Finland, 20/08/18. <https://doi.org/10.4230/LIPIcs.ESA.2018.20>

[Link to publication on Research at Birmingham portal](#)

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Parameterized Approximation Algorithms for Bidirected Steiner Network Problems

Rajesh Chitnis¹

University of Warwick, UK
rajeshchitnis@gmail.com

Andreas Emil Feldmann²

Charles University, Prague, Czechia
feldmann.a.e@gmail.com

Pasin Manurangsi³

University of California, Berkeley, USA
pasin@berkeley.edu

Abstract

The DIRECTED STEINER NETWORK (DSN) problem takes as input a directed edge-weighted graph $G = (V, E)$ and a set $\mathcal{D} \subseteq V \times V$ of k demand pairs. The aim is to compute the cheapest network $N \subseteq G$ for which there is an $s \rightarrow t$ path for each $(s, t) \in \mathcal{D}$. It is known that this problem is notoriously hard as there is no $k^{1/4-o(1)}$ -approximation algorithm under Gap-ETH, even when parameterizing the runtime by k [Dinur & Manurangsi, ITCS 2018]. In light of this, we systematically study several special cases of DSN and determine their parameterized approximability for the parameter k .

For the BI-DSN_{PLANAR} problem, the aim is to compute a planar optimum solution $N \subseteq G$ in a bidirected graph G , i.e. for every edge uv of G the reverse edge vu exists and has the same weight. This problem is a generalization of several well-studied special cases. Our main result is that this problem admits a parameterized approximation scheme (PAS) for k . We also prove that our result is tight in the sense that (a) the runtime of our PAS cannot be significantly improved, and (b) it is unlikely that a PAS exists for any generalization of BI-DSN_{PLANAR}, unless FPT=W[1]. Additionally we study several generalizations of BI-DSN_{PLANAR} and obtain upper and lower bounds on obtainable runtimes parameterized by k .

One important special case of DSN is the STRONGLY CONNECTED STEINER SUBGRAPH (SCSS) problem, for which the solution network $N \subseteq G$ needs to strongly connect a given set of k terminals. It has been observed before that for SCSS a parameterized 2-approximation exists when parameterized by k [Chitnis et al., IPEC 2013]. We show a tight inapproximability result: under Gap-ETH there is no $(2 - \epsilon)$ -approximation algorithm parameterized by k (for any $\epsilon > 0$). To the best of our knowledge, this is the first example of a W[1]-hard problem admitting a non-trivial parameterized approximation factor which is also known to be tight! Additionally we show that when restricting the input of SCSS to bidirected graphs, the problem remains NP-hard but becomes FPT for k .

2012 ACM Subject Classification Theory of computation → Routing and network design problems, Theory of computation → Fixed parameter tractability

Keywords and phrases Directed Steiner Network, Strongly Connected Steiner Subgraph, Parameterized Approximations, Bidirected Graphs, Planar Graphs

¹ Supported by ERC grant 2014-CoG 647557. Part of this work was done while at Weizmann Institute of Science, Israel and supported by Israel Science Foundation grant #897/13

² Supported by the Czech Science Foundation GAČR (grant #17-10090Y), and by the Center for Foundations of Modern Computer Science (Charles Univ. project UNCE/SCI/004).

³ This work was done while the author was visiting Weizmann Institute of Science.



© Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi;
licensed under Creative Commons License CC-BY

26th Annual European Symposium on Algorithms (ESA 2018).

Editors: Yossi Azar, Hannah Bast, and Grzegorz Herman; Article No. 20; pp. 20:1–20:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.20

Related Version A full version of this paper can be found in [12], <https://arxiv.org/abs/1707.06499>.

Acknowledgements We would like to thank Petr Kolman and Hans Raj Tiwary for giving valuable feedback on a draft of this paper.

1 Introduction

In this paper we study the DIRECTED STEINER NETWORK (DSN) problem,⁴ in which a directed edge-weighted graph $G = (V, E)$ is given together with a set of k demands $\mathcal{D} = \{(s_i, t_i)\}_{i=1}^k \subseteq V \times V$. The aim is to compute a minimum cost (in terms of edge weights) network $N \subseteq G$ containing a directed $s_i \rightarrow t_i$ path for each $i \in \{1, \dots, k\}$. This well-studied problem has applications in network design [38], and for instance models the setting where nodes in a radio or ad-hoc wireless network connect to each other unidirectionally [10, 57].

The DSN problem is notoriously hard. First of all, it is NP-hard, and one popular way to handle NP-hard problems is to efficiently compute an α -approximation, i.e., a solution that is guaranteed to be at most a factor α worse than the optimum. For this paradigm we typically demand that the algorithm computing such a solution runs in polynomial time in the input size $n = |V|$. However for DSN it is known that even computing an $O(2^{\log^{1-\varepsilon} n})$ -approximation is not possible [18] in polynomial time, unless $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$. It is possible to obtain approximation factors $O(n^{2/3+\varepsilon})$ and $O(k^{1/2+\varepsilon})$ though [3, 9, 25]. For settings where the number k of demands is fairly small, one may aim for algorithms that only have a mild exponential runtime blow-up in k , i.e., a runtime of the form $f(k) \cdot n^{O(1)}$, where $f(k)$ is some function independent of n . If an algorithm computing the optimum solution with such a runtime exists for a computable function $f(k)$, then the problem is called *fixed-parameter tractable (FPT)* for parameter k . However it is unlikely that DSN is FPT for this well-studied parameter, as it is known to be W[1]-hard [31] for k . In fact one can show [14, 22] that under the *Exponential Time Hypothesis (ETH)* there is no algorithm computing the optimum in time $f(k) \cdot n^{o(k)}$ for any function $f(k)$ independent of n . ETH assumes that there is no $2^{o(n)}$ time algorithm to solve 3SAT [33, 34]. The best we can hope for is therefore a so-called *XP-algorithm* computing the optimum in time $n^{O(k)}$, and this was also shown to exist by Feldman and Ruhl [24].

None of the above algorithms for DSN seem satisfying though, either due to slow runtimes or large approximation factors, and this is hardly surprising given the problem's inherent complexity. To circumvent the hardness of the problem, one may aim for *parameterized approximations*, which have recently received increased attention for various problems (see e.g. [5, 8, 11, 13, 23, 26, 42, 44, 46, 49, 59, 62, 21, 4, 37]). In this paradigm an α -approximation is computed in time $f(k) \cdot n^{O(1)}$ for parameter k , where $f(k)$ again is a computable function independent of n . Unfortunately, a recent result by Dinur and Manurangsi [17]⁵ excludes significant improvements over the known polynomial time approximation algorithms [3, 9, 25], even if allowing a runtime parameterized in k . More specifically, no $k^{1/4-o(1)}$ -approximation

⁴ Also sometimes called DIRECTED STEINER FOREST. Note however that in contrast to the undirected STEINER FOREST problem, an optimum solution to DSN is not necessarily a forest.

⁵ In a previous version of this work, we showed that no $k^{o(1)}$ -approximation is possible for DSN in time $f(k) \cdot n^{O(1)}$. This result is now subsumed by [17].

is possible in time $f(k) \cdot n^{O(1)}$ for any function $f(k)$ under the *Gap Exponential Time Hypothesis* (*Gap-ETH*)⁶, which postulates that there exists a constant $\varepsilon > 0$ such that no (possibly randomized) algorithm running in $2^{o(n)}$ time can distinguish whether it is possible to satisfy all or at most a $(1 - \varepsilon)$ -fraction of clauses of any given 3SAT formula [16, 48].

Given these hardness results, the main question we explore is: what approximation factors and runtimes are possible for special cases of DSN when parametrizing by k ? There are two types of standard special cases that are considered in the literature:

- Restricting the input graph G to some special graph class. A typical assumption for instance is that G is planar.⁷
- Restricting the pattern of the demands in \mathcal{D} . For example, one standard restriction is to have a set $R \subseteq V$ of *terminals*, a fixed *root* $r \in R$, and demand set $\mathcal{D} = \{(r, t) \mid t \in R\}$, which is the well-known DIRECTED STEINER TREE (DST) problem.

In fact, an optimum solution to the DST problem is an arborescence (hence the name), i.e., it is planar. Thus if an algorithm is able to compute (an approximation to) the cheapest planar DSN solution in an otherwise unrestricted graph, it can be used for both the above types of restrictions: it can of course be used if the input graph is planar as well, and it can also be used if the demand pattern implies that the optimum must be planar. Taking the structure of the optimum solution into account has been a fruitful approach leading to several results on related problems, both for approximation and fixed-parameter tractability, from which we also draw some of the inspiration for our results (cf. Section 1.2). A main focus of our work is to systematically explore the influence of the structure of optimum solutions on the complexity of the DSN problem. Formally, fixing a class \mathcal{K} of graphs, we define the $\text{DSN}_{\mathcal{K}}$ problem, which asks for an optimum solution network $N \subseteq G$ for k given demands such that $N \in \mathcal{K}$. The $\text{DSN}_{\mathcal{K}}$ problem has been implicitly studied in several results before for various classes \mathcal{K} , in particular when \mathcal{K} contains either planar graphs, or graphs of bounded treewidth⁸ (cf. Table 1).

Another special case we consider is when the input graph G is *bidirected*, i.e., for every edge uv of G the reverse edge vu exists in G as well and has the same weight as uv . This naturally captures the problem variant between the notoriously hard DSN problem on directed graphs and its undirected counterpart the STEINER FOREST (SF) problem. As the former does not allow any $k^{1/4-o(1)}$ -approximation in time $f(k) \cdot n^{O(1)}$ under Gap-ETH [17], while the latter is FPT [53, 27, 19] for parameter k , it is interesting to ask what happens between these two extremes. Bidirected graphs also model the realistic setting [10, 57, 61, 43] when the cost of transmitting from a node u to a node v in a wireless network is the same in both directions, which for instance happens if the nodes all have the same transmitter model.

We meticulously study several special cases of DSN resulting from the above restrictions, and prove matching upper and lower bounds on runtimes parameterized by k . We now give a brief overview of the studied problems emphasizing the main insights, and refer to Section 1.1 for a detailed exposition of our obtained results.

$\text{BI-DSN}_{\text{PLANAR}}$, i.e., the $\text{DSN}_{\mathcal{K}}$ problem on bidirected inputs, where \mathcal{K} is the class of planar graphs: For this problem we present our main result, which is that $\text{BI-DSN}_{\text{PLANAR}}$ admits a *parameterized approximation scheme (PAS)*, i.e., an algorithm that for any $\varepsilon > 0$ computes

⁶ Gap-ETH follows from ETH given other standard conjectures, such as the existence of linear sized PCPs or exponentially-hard locally-computable one-way functions. See [8, 2] for more details.

⁷ A directed graph is planar if the underlying undirected graph is.

⁸ Here the *undirected* treewidth is meant, i.e., the treewidth of the underlying undirected graph.

a $(1 + \varepsilon)$ -approximation in $f(\varepsilon, k) \cdot n^{g(\varepsilon)}$ time for some functions f and g . We also prove that, unless $\text{FPT} = \text{W}[1]$, no *efficient parameterized approximation scheme (EPAS)* exists, i.e., there is no algorithm computing a $(1 + \varepsilon)$ -approximation in $f(\varepsilon, k) \cdot n^{O(1)}$ time for any function f . Thus the runtime of our algorithm cannot be significantly improved.

BI-DSN, i.e., the DSN problem on bidirected inputs: The above PAS for the restricted $\text{BI-DSN}_{\text{PLANAR}}$ problem begs the question of whether a PAS also exists for any more general problems, such as **BI-DSN**. However we prove that **BI-DSN** does not admit a PAS under Gap-ETH . At the same time it is not too hard to obtain constant approximations in parameterized or polynomial time, given known algorithms for **SF**. When aiming for optimum solutions however, surprisingly we can show that **BI-DSN** is almost as hard as **DSN** (with almost-matching runtime lower bound under ETH). Thus the complexity of the in-between bidirected setting resembles that of the directed setting in terms of FPT algorithms, while in terms of approximations it is more similar to the undirected setting.

Apart from the **DST** problem, another well-studied special case of **DSN** with restricted demands is when the demand pairs form a cycle, i.e., we are given a set $R = \{t_1, \dots, t_k\}$ of k terminals and the set of demands is $\mathcal{D} = \{(t_i, t_{i+1})\}_{i=1}^k$ where $t_{k+1} = t_1$. Since this implies that any optimum solution is strongly connected, this problem is accordingly known as the **STRONGLY CONNECTED STEINER SUBGRAPH (SCSS)** problem. In contrast to **DST**, it is implicit from [31] (by a reduction from the **CLIQUE** problem) that optimum solutions to **SCSS** do not belong to any minor-closed graph class. Thus **SCSS** is not easily captured by some $\text{DSN}_{\mathcal{K}}$ problem for a restricted class \mathcal{K} . Nevertheless it is still possible to exploit the structure of the optimum solution to **SCSS**, which results in the following findings.

SCSS: It is known that a 2-approximation is obtainable [13] when parametrizing by k . We prove that the factor of 2 is best possible under Gap-ETH . To the best of our knowledge, this is the first example of a $\text{W}[1]$ -hard problem having a parameterized approximation algorithm with non-trivial approximation factor (in this case 2), which is also known to be tight!

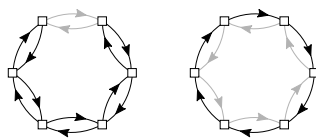
BI-SCSS, i.e., the **SCSS** problem on bidirected inputs: As for **BI-DSN**, one might think that **BI-SCSS** is easily solvable via its undirected version, i.e., the well-known **STEINER TREE (ST)** problem, which is FPT [53, 19] for parameter k . However, it is not the case that simply taking an optimum undirected solution twice in a bidirected graph will produce a (near-)optimum solution to **BI-SCSS** (see Figure 1). Nevertheless we prove that **BI-SCSS** is FPT for parameter k as well, while also being NP-hard . Our algorithm is non-trivial and does not apply any methods used for undirected graphs. To the best of our knowledge, bidirected inputs are the first example where **SCSS** remains NP-hard but turns out to be FPT parameterized by $k!$. Thus in contrast to **BI-DSN**, the complexity of the in-between **BI-SCSS** problem resembles that of the undirected variant (the **ST** problem) rather than the directed version (the **SCSS** problem).

1.1 Our results

Due to space constraints, almost all proofs of the following theorems are deferred to the full version of the paper [12].

Bidirected inputs with planar solutions. Our main theorem implies the existence of a PAS for $\text{BI-DSN}_{\text{PLANAR}}$, where the parameter is the number k of demands.

► **Theorem 1.** *For any $\varepsilon > 0$, there is a $\max \left\{ 2^{k^{2^{O(1/\varepsilon)}}}, n^{2^{O(1/\varepsilon)}} \right\}$ time algorithm for $\text{BI-DSN}_{\text{PLANAR}}$, that computes a $(1 + \varepsilon)$ -approximation.*



■ **Figure 1** A BI-SCSS instance where all vertices are terminals. Left: Black edges show a solution which takes an undirected optimum twice. Right: The actual optimum solution is shown in black.

As $\text{BI-DSN}_{\text{PLANAR}}$ is a rather restricted special case of DSN, one may at this point rightfully ask: Should it not be possible to obtain better runtimes and/or should it not be possible to even compute the optimum solution when parametrizing by k ? And could it not be that a similar result is true in more general settings, when for instance the input is bidirected but the optimum is not restricted to a planar graph? We prove that both questions can be answered in the negative.

First off, it is not hard to prove that a *polynomial time approximation scheme (PTAS)* is not possible for $\text{BI-DSN}_{\text{PLANAR}}$, i.e., it is necessary to parametrize by k in Theorem 1. This is implied by the following result, since (as mentioned before) a PTAS for $\text{BI-DSN}_{\text{PLANAR}}$ would also imply a PTAS for BI-DST, i.e., the DST problem on bidirected input graphs.

► **Theorem 2.** *The BI-DST problem is APX-hard.*

One may wonder however, whether parametrizing by k doesn't make the $\text{BI-DSN}_{\text{PLANAR}}$ problem FPT, so that approximating the planar optimum as in Theorem 1 would in fact be unnecessary. Furthermore, even if it is necessary to approximate, one may ask whether the runtime given in Theorem 1 can be improved. In particular, note that the runtime we obtain in Theorem 1 is similar to that of a PTAS, i.e., the exponent of n in the running time depends on ε . Ideally we would like an EPAS, which has a runtime of the form $f(k, \varepsilon) \cdot n^{O(1)}$, i.e., we would like to treat ε as a parameter as well. The following theorem shows that both approximating and runtime dependence on ε are in fact necessary in Theorem 1.

► **Theorem 3.** *The $\text{BI-DSN}_{\text{PLANAR}}$ problem is $\text{W}[1]$ -hard parameterized by k . Moreover, under ETH, for any computable functions $f(k)$ and $f(k, \varepsilon)$, and parameters k and $\varepsilon > 0$, the $\text{BI-DSN}_{\text{PLANAR}}$ problem has no $f(k) \cdot n^{o(\sqrt{k})}$ time algorithm to compute the optimum solution, and has no $f(k, \varepsilon) \cdot n^{o(\sqrt{k})}$ time algorithm to compute a $(1 + \varepsilon)$ -approximation.*

It stands out that to compute optimum solutions, this theorem rules out runtimes for which the dependence of the exponent of n is $o(\sqrt{k})$, while for the general DSN problem, as mentioned above, the both necessary and sufficient dependence of the exponent is linear in k [24, 14]. Could it be that $\text{BI-DSN}_{\text{PLANAR}}$ is just as hard as DSN when computing optimum solutions? The answer is no, as the next theorem shows.

► **Theorem 4.** *There is a $2^{O(k^{3/2} \log k)} \cdot n^{O(\sqrt{k})}$ time algorithm to compute the optimum solution for $\text{BI-DSN}_{\text{PLANAR}}$.*

This result is an example of the so-called “square-root phenomenon”: planarity often allows runtimes that improve the exponent by a square root factor in terms of the parameter when compared to the general case [28, 50, 40, 47, 41, 52, 55, 54, 51]. Interestingly though, Chitnis et al. [14] show that under ETH, no $f(k) \cdot n^{o(k)}$ time algorithm can compute the optimum solution to $\text{DSN}_{\text{PLANAR}}$. Thus assuming a bidirected input graph in Theorem 4 is necessary (under ETH) to obtain a factor of $O(\sqrt{k})$ in the exponent of n .

Bidirected inputs. Since in contrast to $\text{BI-DSN}_{\text{PLANAR}}$, the BI-DSN problem does not restrict the optimum solutions, one may wonder whether a parameterized approximation scheme as in Theorem 1 is possible for this more general case as well. We answer this in the negative by proving the following result, which implies that restricting the optima to planar graphs was necessary for Theorem 1.

► **Theorem 5.** *Under Gap-ETH, there exists a constant $\alpha > 1$ such that for any computable function $f(k)$ there is no $f(k) \cdot n^{O(1)}$ time algorithm that computes an α -approximation for BI-DSN.*

We leave open whether a similar inapproximability result can be obtained for the other obvious generalization of $\text{BI-DSN}_{\text{PLANAR}}$, in which the input graph is unrestricted but we need to compute the planar optimum, i.e., the $\text{DSN}_{\text{PLANAR}}$ problem. We conjecture that no approximation scheme exists for this problem either.

What approximation factors can be obtained for BI-DSN when parametrizing by k , given the lower bound of Theorem 5 on one hand, and the before-mentioned result [17] that rules out a $k^{1/4-o(1)}$ -approximation for DSN in time parameterized by k on the other? It turns out that it is not too hard to obtain a constant approximation for BI-DSN, given the similarity of bidirected graphs to undirected graphs. In particular, relying on the fact that for the undirected version of DSN, i.e. the SF problem, there is a polynomial time 2-approximation algorithm [1], and an FPT algorithm based on [19], we obtain the following theorem, which is also in contrast to Theorem 2.

► **Theorem 6.** *The BI-DSN problem admits a 4-approximation in polynomial time, and a 2-approximation in $2^{O(k \log k)} \cdot n^{O(1)}$ time.*

Even if Theorem 5 in particular shows that BI-DSN cannot be FPT under Gap-ETH, it does not give a strong lower bound on the runtime dependence in the exponent of n . However using the weaker ETH assumption we can obtain such a lower bound, as the next theorem shows. Interestingly, the obtained lower bound implies that when aiming for optimum solutions, the restriction to bidirected inputs does not make DSN much easier than the general case, as also for BI-DSN the $n^{O(k)}$ time algorithm by [24] is essentially best possible. This is in contrast to the $\text{BI-DSN}_{\text{PLANAR}}$ problem where the square-root phenomenon takes effect as shown by Theorem 4.

► **Theorem 7.** *The BI-DSN problem is $W[1]$ -hard parameterized by k . Moreover, under ETH there is no $f(k) \cdot n^{o(k/\log k)}$ time algorithm for BI-DSN, for any computable function $f(k)$.*

Thus when considering bidirected inputs, which lie between directed and undirected graphs, by Theorem 6 the complexity of the BI-DSN problem rather resembles the undirected variant (the SF problem) in terms of approximations, while by Theorem 7 it resembles the directed version (the DSN problem) in terms of FPT algorithms.

Strongly connected solutions. Just like the more general DSN problem, the SCSS problem is $W[1]$ -hard [31] parameterized by k , and is also hard to approximate as no polynomial time $O(\log^{2-\varepsilon} n)$ -approximation is possible [32], unless $\text{NP} \subseteq \text{ZTIME}(n^{\text{polylog}(n)})$. However it is possible to exploit the structure of the optimum to SCSS to obtain a 2-approximation algorithm parameterized by k , as observed by Chitnis et al. [13]. This is because any strongly connected graph is the union of two arborescences, and these form solutions to DST. The 2-approximation follows, since DST is FPT by the classic result of [19]. Thus in contrast to DSN, for SCSS it is possible to beat any approximation factor obtainable in polynomial time when parametrizing by k .

► **Theorem 8** ([13]). *The SCSS problem admits a 2-approximation in $3^k \cdot n^{O(1)}$ time.*

An obvious question now is whether the approximation ratio of this rather simple algorithm can be improved. Interestingly we are able to show that this is not the case. To the best of our knowledge, this is the first example of a W[1]-hard problem having a parameterized approximation algorithm with non-trivial approximation factor (in this case 2), which is also known to be tight!

► **Theorem 9.** *Under Gap-ETH, for any $\varepsilon > 0$ and any computable function $f(k)$, there is no $f(k) \cdot n^{O(1)}$ time algorithm that computes a $(2 - \varepsilon)$ -approximation for SCSS.*

Bidirected inputs with strongly connected solutions. In light of the above results for restricted cases of DSN, what can be said about restricted cases of SCSS? It is implicit in the work of Chitnis et al. [14] that $\text{SCSS}_{\text{PLANAR}}$, i.e., the problem of computing the optimum strongly connected planar optimum, can be solved in $2^{O(k \log k)} \cdot n^{O(\sqrt{k})}$ time, while under ETH no $f(k) \cdot n^{o(\sqrt{k})}$ time algorithm is possible. Hence $\text{SCSS}_{\text{PLANAR}}$ is slightly easier than $\text{DSN}_{\text{PLANAR}}$ where the exponent of n needs to be linear in k , as mentioned before. On the other hand, the BI-SCSS problem turns out to be a lot easier to solve than BI-DSN. This is implied by the next theorem, which stands in contrast to Theorem 5 and Theorem 7. In particular, the in-between BI-SCSS problem behaves more like the undirected ST problem than the directed SCSS problem.

► **Theorem 10.** *There is a $2^{O(2^{k^2-k})} \cdot n^{O(1)}$ time algorithm for BI-SCSS, i.e., it is FPT for parameter k .*

Could it be that BI-SCSS is even solvable in polynomial time? We prove that this is not the case, as it is NP-hard. To the best of our knowledge, the class of bidirected graphs is the first example where SCSS remains NP-hard but turns out to be FPT parameterized by k ! Moreover, note that the above algorithm has a doubly exponential runtime in k^2 . We conjecture that a single exponential runtime should suffice, and we also obtain a lower bound result of this form, even if we restrict the optimum solutions to very simple planar graphs, namely cycles.

► **Theorem 11.** *The $\text{BI-SCSS}_{\text{CYCLE}}$ problem is NP-hard. Moreover, under ETH there is no $2^{o(k)} \cdot n^{O(1)}$ time algorithm for $\text{BI-SCSS}_{\text{CYCLE}}$.*

► **Remark.** For ease of notation, throughout this paper we chose to use the number of demands k uniformly as the parameter. Alternatively one might also consider the smaller parameter $|R|$, where $R = \bigcup_{i=1}^k \{s_i, t_i\}$ is the set of terminals. Note for instance that in case of the SCSS problem, $k = |R|$, while for DSN, k can be as large as $\Theta(|R|^2)$ (cf. [22]). However we always have $k \geq |R|/2$, since the demands can form a matching in the worst case. It is interesting to note that all our algorithms for DSN have the same running time for parameter $|R|$ as for parameter k . That is, we may set $k = |R|$ in Theorem 1, 4, and 6.

1.2 Our techniques

It is already apparent from the above exposition of our results, that understanding the structure of the optimum solution is a powerful tool when studying DSN and its related problems (see Table 1). This is also apparent when reading the literature on these problems, and we draw some of our inspiration from these known results, as described below.

■ **Table 1** Summary of achievable runtimes for DSN and SCSS when parameterizing by k . Some of the previous results are implicit and, in the papers, are rather stated for the case when the input graphs are restricted to the same class as the optimum solutions. Non-bracketed reference numbers refer to theorems of this paper.

problem	algorithms			lower bounds		
	approx.	runtime	ref.	approx.	runtime	ref.
DSN	–	$n^{O(k)}$	[24]	–	$f(k) \cdot n^{o(k)}$	[31]
DSN	$O(k^{\frac{1}{2}+\varepsilon})$	$n^{O(1)}$	[9]	$k^{\frac{1}{4}-o(1)}$	$f(k) \cdot n^{O(1)}$	[17]
DSN _{TW: ω}	–	$2^{O(k\omega \log \omega)} \cdot n^{O(\omega)}$	[27]	–	$f(k, \omega) \cdot n^{o(\omega)}$	[27]
BI-DSN _{PLANAR}	$1 + \varepsilon$	$\max\{2^{k^{2^{O(1/\varepsilon)}}}, n^{2^{O(1/\varepsilon)}}\}$	1	$1 + \varepsilon$	$f(\varepsilon, k) \cdot n^{o(\sqrt{k})}$	3
BI-DSN _{PLANAR}	–	$2^{O(k^{3/2} \log k)} \cdot n^{O(\sqrt{k})}$	4	–	$f(k) \cdot n^{o(\sqrt{k})}$	3
DSN _{PLANAR}	–	$n^{O(k)}$	[24]	–	$f(k) \cdot n^{o(k)}$	[14]
BI-DSN	–	$n^{O(k)}$	[24]	–	$f(k) \cdot n^{o(k/\log k)}$	7
BI-DSN	2	$2^{O(k \log k)} \cdot n^{O(1)}$	6	$\alpha \in \Theta(1)$	$f(k) \cdot n^{O(1)}$	5
BI-DSN	4	$n^{O(1)}$	6	$\alpha \in \Theta(1)$	$n^{O(1)}$	2
SCSS	–	$n^{O(k)}$	[24]	–	$f(k) \cdot n^{o(k/\log k)}$	[14]
SCSS	2	$3^k \cdot n^{O(1)}$	[13]	$2 - \varepsilon$	$f(k) \cdot n^{O(1)}$	9
SCSS _{PLANAR}	–	$2^{O(k)} \cdot n^{O(\sqrt{k})}$	[14]	–	$f(k) \cdot n^{o(\sqrt{k})}$	[14]
BI-SCSS	–	$2^{O(2^{k^2-k})} \cdot n^{O(1)}$	10	–	$2^{o(k)} \cdot n^{O(1)}$	11

For our approximation scheme for BI-DSN_{PLANAR}, we generalize the insights on the structure of optimum solutions to the classical STEINER TREE (ST) problem for our main result in Theorem 1. For the ST problem, an *undirected* edge-weighted graph is given together with a terminal set R , and the task is to compute the cheapest tree connecting all k terminals. For the ST problem only polynomial time 2-approximations were known [30, 60], until it was taken into account [36, 56, 63, 58] that any optimum Steiner tree can be decomposed into so-called *full components*, i.e., subtrees for which exactly the leaves are terminals. If a full component contains only a small subset of size k' of the terminals, it is the solution to an ST instance, for which the optimum can be computed efficiently in time $(2 + \delta)^{k'} \cdot n^{O(1)}$ for any constant $\delta > 0$ using the algorithm of Mölle et al. [53]. A fundamental observation proved by Borchers and Du [6] is that for any k' there exists a solution to ST of cost at most $1 + \frac{1}{\lfloor \log_2 k' \rfloor}$ times the optimum, in which every full component contains at most k' terminals. Thus setting $k' = 2^{1/\varepsilon}$ for some constant $\varepsilon > 0$, all full-components with at most $2^{1/\varepsilon}$ terminals can be computed in polynomial time, and among them exists a collection forming a $(1 + \varepsilon)$ -approximation. The key to obtain approximation ratios smaller than 2 for ST is to cleverly select a good subset of all computed full-components. This is for instance done in [7] via an iterative rounding procedure, resulting in an approximation ratio of $\ln(4) + \varepsilon < 1.39$, which currently is the best one known.

Our main technical contribution is to generalize the Borchers-Du [6] Theorem to the BI-DSN_{PLANAR} problem. In particular, to obtain our approximation scheme of Theorem 1, we employ a similar approach by decomposing a BI-DSN_{PLANAR} solution into sub-instances, each containing a small number of terminals. As BI-DSN_{PLANAR} is W[1]-hard by Theorem 3, we cannot hope to compute optimum solutions to each sub-instance as efficiently as for ST.

However, we provide an XP-algorithm with runtime $2^{O(k^{3/2} \log k)} \cdot n^{O(\sqrt{k})}$ for $\text{BI-DSN}_{\text{PLANAR}}$ in Theorem 4. Thus if every sub-instance contains at most $2^{1/\varepsilon}$ terminals, each can be solved in $n^{2^{O(1/\varepsilon)}}$ time, and this accounts for the “non-efficient” runtime of our approximation scheme. Since we allow runtimes parameterized by k , we can then exhaustively search for a good subset of precomputed small optimum solutions to obtain a solution to the given demand set \mathcal{D} . For the latter solution to be a $(1 + \varepsilon)$ -approximation however, we need to generalize the Borchers-Du [6] Theorem for ST to $\text{BI-DSN}_{\text{PLANAR}}$ (see Theorem 13 for the formal statement). This constitutes the bulk of the work to prove Theorem 1.

For our exact algorithms for $\text{BI-DSN}_{\text{PLANAR}}$ and BI-SCSS , we note that also from a parameterized point of view, understanding the structure of the optimum solution to DSN has lead to useful insights in the past. We will leverage one such recent result by Feldmann and Marx [27]. In [27] the above mentioned standard special case of restricting the patterns of the demands in \mathcal{D} is studied in depth. The result is a complete dichotomy over which classes of restricted patterns define special cases of DSN that are FPT and which are $\text{W}[1]$ -hard for parameter k . The high-level idea is that whenever the demand patterns imply optimum solutions of constant treewidth, there is an FPT algorithm computing such an optimum. In contrast, the problem is $\text{W}[1]$ -hard whenever the demand patterns imply the existence of optimum solutions of arbitrarily large treewidth. The FPT algorithm from [27] lies at the heart of all our positive results, and therefore shows that the techniques developed in [27] to optimally solve special cases of DSN can be extended to find (near-)optimum solutions for other $\text{W}[1]$ -hard special cases as well. It is important to note that the algorithm of [27] can also be used to compute the cheapest solution of treewidth at most ω , even if there is an even better solution of treewidth larger than ω (which might be hard to compute). Formally, the result leveraged in this paper is the following.

► **Theorem 12** (implicit in Theorem 5 of [27]). *If \mathcal{K} is the class of graphs with treewidth at most ω , then the $\text{DSN}_{\mathcal{K}}$ problem can be solved in time $2^{O(k\omega \log \omega)} \cdot n^{O(\omega)}$.*

We exploit the algorithm given in Theorem 12 to prove our algorithmic results of Theorem 4 and Theorem 10. In particular, we prove that any $\text{BI-DSN}_{\text{PLANAR}}$ solution has treewidth $O(\sqrt{k})$, from which Theorem 4 follows immediately. For BI-SCSS however, we give an example of an optimum solution of treewidth $\Omega(k)$. Hence we cannot exploit the algorithm of Theorem 12 directly to obtain Theorem 10. In fact on general input graphs, a treewidth of $\Omega(k)$ would imply that the problem is $\text{W}[1]$ -hard by the hardness results in [27] (which was indeed originally shown by Guo et al. [31]). As this stands in stark contrast to Theorem 10, it is particularly interesting that the problem on bidirected input graphs is FPT. We prove this result by decomposing an optimum solution to BI-SCSS into instances of $\text{BI-SCSS}_{\mathcal{K}}$, where \mathcal{K} is the class of directed graphs of treewidth 1 (so-called *poly-trees*). For each such sub-instance we can compute a solution in $2^{O(k)} \cdot n^{O(1)}$ time by using Theorem 12 (for $\omega = 1$), and then combine them into an optimum solution to BI-SCSS .

Our hardness proofs for BI-DSN are based on reductions from the GRID TILING problem [15]. This problem is particularly suited to prove hardness for problems on planar graphs, due to its grid-like structure. We first develop a specific gadget that can be exploited to show hardness for bidirected graphs. This gadget however is not planar. We only exploit the structure of GRID TILING to show that the optimum solution is planar for Theorem 3. For Theorem 7 we modify this reduction to obtain a stronger runtime lower bound, but in the process we lose the property that the optimum is planar.

Our parameterized inapproximability result for SCSS is proved by combining a variant of a known reduction by Guo et al. [31] with a recent parameterized hardness of approximation

result for DENSEST k -SUBGRAPH [8]. Our inapproximability result for BI-DSN is shown by combining our W[1]-hardness reduction with the same hardness of approximation result of DENSEST k -SUBGRAPH.

2 An approximation scheme for $\text{BI-DSN}_{\text{PLANAR}}$

In this section we prove Theorem 1. Note that since we have k demand pairs, it follows that the number of terminals $|R|$ is at most $2k$, where $R = \bigcup_{i=1}^k \{s_i, t_i\}$. Henceforth in this section, we use the upper bound $2k$ on the number of terminals $|R|$ for ease of presentation (when instead we could replace k by $|R|$ in the running time of Theorem 1). The bulk of the proof is captured by the following result, which generalizes the corresponding theorem by Borchers and Du [6] for the ST problem, and which is our main technical contribution. In order to facilitate the definition of a sub-instance to DSN, we encode the demands of a DSN instance using a *pattern graph* H , as also done in [27]: the vertex set of H is the terminal set R , and H contains the directed edge st if and only if (s, t) is a demand. Hence the DSN problem asks for a minimum cost network $N \subseteq G$ having an $s \rightarrow t$ path for each edge st of H .

► **Theorem 13.** *Let G be a bidirected graph, and H a pattern graph on $R \subseteq V(G)$. Let $N \subseteq G$ be an optimum $\text{BI-DSN}_{\text{PLANAR}}$ solution to H , i.e. N is planar. For any $\varepsilon > 0$, there exists a set of patterns \mathcal{H} such that for each $H' \in \mathcal{H}$ there is a feasible $\text{BI-DSN}_{\text{PLANAR}}$ solution $N_{H'} \subseteq G$ and $|V(H')| \leq 2^{O(1/\varepsilon)}$. Furthermore, the union $\bigcup_{H' \in \mathcal{H}} N_{H'}$ of these solutions forms a feasible $\text{BI-DSN}_{\text{PLANAR}}$ solution to H with $\sum_{H' \in \mathcal{H}} \text{cost}(N_{H'}) \leq (1 + \varepsilon) \cdot \text{cost}(N)$.*

Based on Theorem 13 our $(1 + \varepsilon)$ -approximation algorithm proceeds as follows. The first step is to compute an optimum solution for every possible pattern graph on at most $g(\varepsilon) = 2^{O(1/\varepsilon)}$ terminals. Since any pattern graph has at most $2^{\binom{g(\varepsilon)}{2}} < g(\varepsilon)^2$ edges, and there is a total of $2^{\binom{2k}{2}} < 8k^2$ possible demands between the $2k$ terminals, the total number of pattern graphs is $O(k^{2g(\varepsilon)^2}) = k^{2^{O(1/\varepsilon)}}$. For each pattern the algorithm computes the optimum $\text{BI-DSN}_{\text{PLANAR}}$ solution in time $2^{g(\varepsilon)^{3/2} \log g(\varepsilon)} \cdot n^{O(\sqrt{g(\varepsilon)})} = n^{2^{O(1/\varepsilon)}}$ using the algorithm of Theorem 4. This amounts to a total runtime of $k^{2^{O(1/\varepsilon)}} \cdot n^{2^{O(1/\varepsilon)}}$ up to this point. The algorithm then proceeds by considering each subset \mathcal{H} of the pattern graphs, and checking whether the union of the precomputed optimum solutions to all $H' \in \mathcal{H}$ forms a feasible solution to the input pattern H on R . As there are $2^{O(k^{2g(\varepsilon)^2})}$ subsets \mathcal{H} , and checking whether a subset induces a feasible solution can be done in polynomial time, this takes $2^{O(k^{2g(\varepsilon)^2})} \cdot n^{O(1)} = 2^{k^{2^{O(1/\varepsilon)}}} \cdot n^{O(1)}$ time. Among all feasible unions the algorithm outputs the solution with smallest cost. According to Theorem 13 this solution is a $(1 + \varepsilon)$ -approximation, and the total runtime is $k^{2^{O(1/\varepsilon)}} \cdot n^{2^{O(1/\varepsilon)}} + 2^{k^{2^{O(1/\varepsilon)}}} \cdot n^{O(1)} = \max \left\{ 2^{k^{2^{O(1/\varepsilon)}}}, n^{2^{O(1/\varepsilon)}} \right\}$. Thus we obtain Theorem 1.

Note that even though the output of the algorithm is a $(1 + \varepsilon)$ -approximation to the optimum $\text{BI-DSN}_{\text{PLANAR}}$ solution, the computed solution may not be planar, as it is the union of several planar graphs. Theorem 13 shows though that the structure of the optimum can be exploited to compute a near-optimum solution. We also note that the Borchers-Du[6] Theorem for the ST problem implies the existence of a *polynomial-sized* $(1 + \varepsilon)$ -approximate kernel for ST, as recently shown by Lokshtanov et al. [46]. By the same arguments this is also true for $\text{BI-DSN}_{\text{PLANAR}}$, due to Theorem 13. We refer to [46] for more details.

► **Corollary 14** (cf. [46]). *The $\text{BI-DSN}_{\text{PLANAR}}$ problem admits a polynomial-size approximate kernelization scheme (PSAKS) parameterized by k .*

It remains to prove Theorem 13. For this we assume we know the optimum planar solution $N \subseteq G$, and first use a standard transformation on N , so that each terminal has only 1 neighbour, each Steiner vertex has exactly 3 neighbours, and every pair of edges uv and vu have unique costs. Furthermore, let G_N be the graph spanned by the edge set $\{uv, vu \in E(G) \mid uv \in E(N)\}$, i.e. it is the underlying bidirected graph of N after performing the transformations on N . In particular, also in G_N each terminal has only 1 neighbour, each Steiner vertex has exactly 3 neighbours, and every pair of edges uv and vu have unique costs. It is not hard to see that proving Theorem 13 for the obtained optimum solution N in G_N implies the same result for the original optimum solution in G , by reversing all transformations.

The proof consists of two parts, of which the first exploits the bidirectedness of G_N , while the second exploits that the optimum N is planar. The first part will identify paths connecting each Steiner vertex to some terminal in such a way that the paths do not overlap much. This will enable us to select a subset of these paths in the second part, so that the total weight of the selected paths is an ε -fraction of the cost of the optimum solution. This subset of paths will be used to connect terminals to the boundary vertices of small regions into which we divide the optimum. These regions extended by the paths then form solutions to sub-instances to DSN, which together have a cost of $1 + \varepsilon$ times the optimum. The first part is captured by the next lemma, where $\text{cost}(G')$ denotes the total edge weight of a graph G' .

► **Lemma 15.** *Let $N \subseteq G_N$ be the optimum $\text{BI-DSN}_{\text{PLANAR}}$ solution to a pattern graph H on $R \subseteq V(G_N)$. For every Steiner vertex $v \in V(N) \setminus R$ of N there is a path P_v in G_N , such that P_v is a $v \rightarrow t$ path to some terminal $t \in R$, and the total cost $\sum_{v \in V(N) \setminus R} \text{cost}(P_v)$ of these paths is $O(\text{cost}(N))$.*

For the second part we give each vertex v of N a weight $c(v)$, which is zero for terminals and equal to $\text{cost}(P_v)$ for each Steiner vertex $v \in V(N) \setminus R$ and corresponding path P_v given by Lemma 15. We now divide the optimum solution N into regions of small size, such that the boundaries of the regions have small total weight. Formally, a *region* is a subgraph of N , and an *r -division* is given by a partition of the edges of N , each spanning a region with at most r vertices. A *boundary vertex* of an r -division is a vertex that lies in at least two regions. In a *weak r -division*, as for instance defined in [35], we bound the total number of boundary vertices and the number of regions (it is called “weak” since it does not bound the boundary vertices of each region individually). For unweighted planar graphs it can be shown that there is an r -division with only $O(n/\sqrt{r})$ boundary vertices and $O(n/r)$ regions [35, 29]. To prove this, a separator theorem is applied recursively until each resulting region is small enough. The bound on the number of boundary vertices follows from the well-known fact that any planar graph has a small separator of size $O(\sqrt{n})$.

We however need to bound the total weight of the boundary vertices, i.e. we need a *weighted weak r -division*. Unfortunately, separator theorems are not helpful here, since they only bound the number of vertices in the separator but cannot bound their weight. Instead we leverage techniques developed for the Klein-Plotkin-Rao (KPR) Theorem [45, 39] in order to show that there is an r -division for which the total weight of all boundary vertices is an $O(1/\log r)$ -fraction of the total weight $\sum_{v \in V(N)} c(v)$, if the graph has constant maximum degree. We later set $r = 2^{1/\varepsilon}$ in order to obtain an ε -fraction of the total weight. Even though the obtained fraction is exponentially worse than the $O(1/\sqrt{r})$ -fraction for unweighted graphs obtained in [35, 29], it follows from a lower bound result of Borchers and Du [6] that for weighted graphs this is best possible, even if the graph is a tree. In contrast to the

unweighted case, we also do not guarantee any bound on the number of regions, and we do not need such a bound either. Our proof follows the outlines of the proof given by Lee [45] for the KPR Theorem. In the following, $c(S) = \sum_{v \in S} c(v)$ for any set of vertices S .

► **Lemma 16.** *Let N be a directed planar graph for which each vertex has at most 3 neighbours, and let each vertex v of N have a weight $c(v) \in \mathbb{R}$. For any $r \in \mathbb{N}$ there is a partition \mathcal{E} of the edges of N for which every set in \mathcal{E} spans at most r vertices, and if B is the set of boundary vertices of the regions spanned by the sets in \mathcal{E} , then $c(B) = O\left(\frac{c(V(N))}{\log r}\right)$.*

We here only prove some parts of Lemma 15 (cf. [12] for the full version of the paper).

Proof of Lemma 15. We begin by analysing the structure of optimal DSN solutions in bidirected graphs. Here a *condensation graph* of a directed graph results from contracting each strongly connected component, which hence is a DAG. A *poly-forest* is obtained by directing the edges of an undirected forest.

► **Claim 17.** *For any solution $N \subseteq G_N$ to a pattern H , there is a solution $M \subseteq G_N$ to H with $\text{cost}(M) \leq \text{cost}(N)$, such that the condensation graph of M is a poly-forest.*

By Claim 17 we may assume w.l.o.g. that the condensation graph of the optimum solution N is a poly-forest. Consider a weakly connected component C of N , i.e. inducing a connected component of the underlying undirected graph \vec{N} . We first extend C to a strongly connected graph C' as follows. Let F be the edges of C that do not lie in a strongly connected component, i.e. they are the edges of the condensation graph of C . Let $\tilde{F} = \{uv \mid vu \in F\}$ be the set containing the reverse edges of F , and let C' be the strongly connected graph spanned by all edges of C in addition to the edges in \tilde{F} . Note that adding \tilde{F} to C increases the cost by at most a factor of two as G_N is bidirected, and the number of neighbours of any vertex does not change. We claim that in fact C' is a *minimal* SCSS solution to the terminal set $R_C \subseteq R$ contained in C , that is, removing any edge of C' will disconnect some terminal pair of R_C .

For this, consider any $s \rightarrow t$ path of C' containing an edge $e \in \tilde{F}$ for some terminal pair $s, t \in R_C$. As the edges F of the condensation graph of C form a poly-tree, every path from s to t in C' must pass through e . In particular there is no $s \rightarrow t$ path in C , and thus there is no edge st in the pattern graph H . Or conversely, for any terminal pair $s, t \in R_C$ for which there is a demand $st \in E(H)$, no $s \rightarrow t$ path in C' passes through an edge of \tilde{F} . Thus the set of paths from s to t is the same in C' and C . Since every edge e of the weakly connected component C is necessary for some such pair $s, t \in R_C$ with $st \in E(H)$, the edge e is still necessary in C' . Moreover, for any of the added edges $uv \in \tilde{F}$ the reverse edge $vu \in F$ was necessary in C to connect some $s \in R_C$ to some $t \in R_C$. As observed above, uv is necessary to connect t to s in C' , since the edges F of the condensation graph form a poly-tree.

As C' is a minimal SCSS solution to the terminals R_C contained within, it is the union of an in-arborescence A_{in} and out-arborescence A_{out} , both with the same root $r \in R_C$ and leaf set $R_C \setminus \{r\}$, since every terminal only has one neighbour in G_N . A *branching point* of an arborescence A is a vertex with at least two children in A . We let $W \subseteq V(C')$ be the set consisting of all terminals R_C and all branching points of A_{in} and A_{out} . We will need that any vertex of C' has a vertex of W in its close vicinity. That is, if $\Delta[v] = \{u \in V(C') \mid u = v \vee uv \in E(C') \vee vu \in E(C')\}$ denotes the inclusive neighbourhood of a vertex v ignoring directions of edges and $\Delta^2[v] = \bigcup_{u \in \Delta[v]} \Delta[u]$, we prove the following.

► **Claim 18.** *For every vertex v of C' , there is a vertex of W in $\Delta^2[v]$.*

As the graph G_N is bidirected, for any v - u path P in the underlying undirected graph \vec{G}_N of G_N , there exists a corresponding directed $v \rightarrow u$ path in G_N of the same cost. Therefore, we can ignore the directions of the edges in C' and the arborescences A_{out} and A_{in} to identify the paths P_v for Steiner vertices v of N . Thus we will only consider paths in the underlying undirected graphs \vec{C}' , \vec{A}_{out} , and \vec{A}_{in} from now on. In particular, we exploit the following observation found in [20] (and also used by [6]) on undirected trees.

► **Claim 19** ([20, Lemma 3.2]). *For any undirected tree T we can find a path $P_v \subseteq T$ for every branching point v , such that P_v leads from v to some leaf of T , and all these paths P_v are pairwise edge-disjoint.*

If a Steiner vertex v of C' is a branching point of A_{out} (A_{in}), we let P_v be the corresponding path in \vec{A}_{out} (\vec{A}_{in}) given by Claim 19 from v to some leaf of A_{out} (A_{in}), which is a terminal. Note that paths in \vec{A}_{in} may overlap with paths in \vec{A}_{out} . However any edge in the union of all the paths P_v chosen so far is contained in at most two such paths, one for a branching point of A_{out} and one for a branching point of A_{in} .

It remains to choose a path P_v for every Steiner vertex v that is neither a branching point of A_{out} nor of A_{in} , i.e. for every vertex not in W . By Claim 18 for any such vertex $v \notin W$ there is a vertex $u \in \Delta^2[v]$ for which $u \in W$. If u is a terminal, then the path P_v is simply the edge vu if $u \in \Delta[v]$ or the corresponding path vwu for some $w \in \Delta[v]$ otherwise. If u is not a terminal but a branching point of A_{out} or A_{in} , then we chose a path P_u for u above. In this case, P_v is the path contained in the walk given by extending the path P_u by the edge vu or the path vwu , respectively. Note that, as any vertex of C' has at most three neighbours, any terminal or branching point $u \in W$ can be used in this way for some vertex $v \notin W$ at most nine times. Therefore any edge in the union of all chosen paths is contained in $O(1)$ paths. Consequently the total cost $\sum_{v \in V(N) \setminus R} \text{cost}(P_v)$ is $O(\text{cost}(C'))$, and as $\text{cost}(C') \leq 2 \text{cost}(C)$ we also get $\sum_{v \in V(N) \setminus R} \text{cost}(P_v) = O(\text{cost}(C))$.

We may repeat these arguments for every weakly connected component of N to obtain the lemma. ◀

References

- 1 Ajit Agrawal, Philip Klein, and R Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing*, 24(3):440–456, 1995.
- 2 Benny Applebaum. Exponentially-Hard Gap-CSP and Local PRG via Local Hardcore Functions. In *FOCS 2017*, pages 836–847, 2017. doi:10.1109/FOCS.2017.82.
- 3 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and directed steiner forest. *Information and Computation*, 222:93–107, 2013.
- 4 Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. Parameterized Intractability of Even Set and Shortest Vector Problem from Gap-ETH. *To appear in ICALP 2018*, 2018. arXiv:1803.09717.
- 5 Edouard Bonnet, Bruno Escoffier, EunJung Kim, and Vangelis T. Paschos. On subexponential and FPT-time inapproximability. In *IPEC*, pages 54–65, 2013.
- 6 Al Borchers and Ding-Zhu Du. The k -Steiner Ratio in Graphs. *SIAM Journal on Computing*, 26(3):857–869, 1997.
- 7 Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *Journal of the ACM*, 60(1):6, 2013.

- 8 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From Gap-ETH to FPT-Inapproximability: Clique, Dominating Set, and More. In *To appear in FOCS*, 2017.
- 9 Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed Steiner network problem. *ACM Transactions on Algorithms*, 7(2):18, 2011.
- 10 W-T Chen and N-F Huang. The strongly connecting problem on multihop packet radio networks. *IEEE Transactions on Communications*, 37(3):293–295, 1989.
- 11 Yijia Chen and Bingkai Lin. The constant inapproximability of the parameterized dominating set problem. In *FOCS*, pages 505–514, 2016.
- 12 Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. Parameterized Approximation Algorithms for Directed Steiner Network Problems. *CoRR*, abs/1707.06499, 2017. [arXiv:1707.06499](#).
- 13 Rajesh Chitnis, MohammadTaghi Hajiaghayi, and Guy Kortsarz. Fixed-parameter and approximation algorithms: A new look. In *IPEC*, pages 110–122, 2013.
- 14 Rajesh Chitnis, MohammadTaghi Hajiaghayi, and Dániel Marx. Tight bounds for planar strongly connected Steiner subgraph with fixed number of terminals (and extensions). In *SODA*, pages 1782–1801, 2014.
- 15 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 16 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016.
- 17 Irit Dinur and Pasin Manurangsi. ETH-Hardness of Approximating 2-CSPs and Directed Steiner Network. In *ITCS*, pages 36:1–36:20, 2018. doi:10.4230/LIPIcs.ITCS.2018.36.
- 18 Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *STOC 1999*, pages 750–759, 1999.
- 19 S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1971.
- 20 Ding-Zhu Du, Yanjun Zhang, and Qing Feng. On better heuristic for Euclidean Steiner minimum trees. In *FOCS 1991*, pages 431–439, 1991.
- 21 Pavel Dvořák, Andreas Emil Feldmann, Dusan Knop, Tomáš Masarík, Tomas Toufar, and Pavel Veselý. Parameterized approximation schemes for steiner trees with small number of steiner vertices. In *STACS*, pages 26:1–26:15, 2018.
- 22 Eduard Eiben, Dušan Knop, Fahad Panolan, and Ondřej Suchý. Complexity of the steiner network problem with respect to the number of terminals. *arXiv preprint*, 2018. [arXiv:1802.08189](#).
- 23 Eduard Eiben, Mithilesh Kumar, Amer E Mouawad, and Fahad Panolan. Lossy kernels for connected dominating set on sparse graphs. *arXiv preprint*, 2017. [arXiv:1706.09339](#).
- 24 Jon Feldman and Matthias Ruhl. The directed Steiner network problem is tractable for a constant number of terminals. *SIAM J. Comput.*, 36(2):543–561, 2006.
- 25 Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved approximation algorithms for directed steiner forest. *J. Comput. Syst. Sci.*, 78(1):279–292, 2012. doi:10.1016/j.jcss.2011.05.009.
- 26 Andreas Emil Feldmann. Fixed parameter approximations for k-center problems in low highway dimension graphs. In *ICALP*, pages 588–600, 2015. doi:10.1007/978-3-662-47666-6_47.
- 27 Andreas Emil Feldmann and Dániel Marx. The complexity landscape of fixed-parameter directed steiner network problems. *CoRR*, abs/1707.06808, 2017. [arXiv:1707.06808](#).

- 28 Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. Subexponential Parameterized Algorithms for Planar and Apex-Minor-Free Graphs via Low Treewidth Pattern Covering. In *FOCS*, pages 515–524, 2016. doi:10.1109/FOCS.2016.62.
- 29 Greg N Frederickson and Joseph Ja’Ja’. Approximation algorithms for several graph augmentation problems. *SIAM Journal on Computing*, 10(2):270–283, 1981.
- 30 E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.
- 31 Jiong Guo, Rolf Niedermeier, and Ondrej Suchý. Parameterized complexity of arc-weighted directed Steiner problems. *SIAM J. Discrete Math.*, 25(2):583–599, 2011.
- 32 Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. In *STOC*, pages 585–594, 2003.
- 33 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k -SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.
- 34 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
- 35 Giuseppe F Italiano, Yahav Nussbaum, Piotr Sankowski, and Christian Wulff-Nilsen. Improved algorithms for min cut and max flow in undirected planar graphs. In *STOC 2011*, pages 313–322, 2011.
- 36 Marek Karpinski and Alexander Zelikovskiy. New approximation algorithms for the Steiner tree problem. *Journal of Combinatorial Optimization*, 1(1):47–65, 1997.
- 37 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the Parameterized Complexity of Approximating Dominating Set. To appear in *STOC 2018*, 2017. arXiv:1711.11029.
- 38 Hervé Kerivin and A Ridha Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.
- 39 Philip Klein, Serge A. Plotkin, and Satish Rao. Excluded Minors, Network Decomposition, and Multicommodity Flow. In *STOC 1993*, pages 682–690, 1993.
- 40 Philip N. Klein and Dániel Marx. Solving Planar k -Terminal Cut in $O(n^{c\sqrt{k}})$ Time. In *ICALP*, pages 569–580, 2012. doi:10.1007/978-3-642-31594-7_48.
- 41 Philip N. Klein and Dániel Marx. A subexponential parameterized algorithm for Subset TSP on planar graphs. In *SODA*, pages 1812–1830, 2014. doi:10.1137/1.9781611973402.131.
- 42 R. Krithika, Pranabendu Misra, Ashutosh Rai, and Prafullkumar Tale. Lossy kernels for graph contraction problems. In *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016)*, volume 65, pages 23:1–23:14, Dagstuhl, Germany, 2016. doi:10.4230/LIPIcs.FSTTCS.2016.23.
- 43 Nhat X Lam, Trac N Nguyen, Min Kyung An, and Dung T Huynh. Dual power assignment optimization and fault tolerance in WSNs. *Journal of Combinatorial Optimization*, 30(1):120–138, 2015.
- 44 Michael Lampis. Parameterized approximation schemes using graph widths. In *ICALP*, pages 775–786, 2014.
- 45 James Lee. A simpler proof of the KPR theorem, 2012. URL: <https://tcsmath.org/2012/01/11/a-simpler-proof-of-the-kpr-theorem/>.
- 46 Daniel Lokshtanov, Fahad Panolan, MS Ramanujan, and Saket Saurabh. Lossy Kernelization. In *STOC*, pages 224–237, 2017.
- 47 Daniel Lokshtanov, Saket Saurabh, and Magnus Wahlström. Subexponential Parameterized Odd Cycle Transversal on Planar Graphs. In *FSTTCS*, pages 424–434, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.424.
- 48 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In *ICALP*, pages 78:1–78:15, 2017.

- 49 Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.
- 50 Dániel Marx. A Tight Lower Bound for Planar Multiway Cut with Fixed Number of Terminals. In *ICALP*, pages 677–688, 2012. doi:10.1007/978-3-642-31594-7_57.
- 51 Dániel Marx, Marcin Pilipczuk, and Michał Pilipczuk. On subexponential parameterized algorithms for Steiner Tree and Directed Subset TSP on planar graphs. *arXiv preprint arXiv:1707.1707.02190*, 2017.
- 52 Dániel Marx and Michał Pilipczuk. Optimal Parameterized Algorithms for Planar Facility Location Problems Using Voronoi Diagrams. In *ESA*, pages 865–877, 2015. doi:10.1007/978-3-662-48350-3_72.
- 53 Daniel Mölle, Stefan Richter, and Peter Rossmanith. A faster algorithm for the steiner tree problem. In *STACS*, pages 561–570, 2006.
- 54 Marcin Pilipczuk, Michał Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Subexponential-Time Parameterized Algorithm for Steiner Tree on Planar Graphs. In *STACS*, pages 353–364, 2013. doi:10.4230/LIPIcs.STACS.2013.353.
- 55 Marcin Pilipczuk, Michał Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Network Sparsification for Steiner Problems on Planar and Bounded-Genus Graphs. In *FOCS*, pages 276–285, 2014. doi:10.1109/FOCS.2014.37.
- 56 Hans Jürgen Prömel and Angelika Steger. A new approximation algorithm for the Steiner tree problem with performance ratio $5/3$. *Journal of Algorithms*, 36:89–101, 2000.
- 57 Ram Ramanathan and Regina Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM*, volume 2, pages 404–413, 2000.
- 58 Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122–134, 2005.
- 59 Sebastian Siebertz. Lossy kernels for connected distance- r domination on nowhere dense graph classes. *arXiv preprint*, 2017. arXiv:1707.09819.
- 60 Vijay Virkumar Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.
- 61 Chen Wang, Myung-Ah Park, James Willson, Yongxi Cheng, Andras Farago, and Weili Wu. On approximate optimal dual power assignment for biconnectivity and edge-biconnectivity. *Theoretical Computer Science*, 396(1-3):180–190, 2008.
- 62 Andreas Wiese. A $(1 + \epsilon)$ -approximation for unsplittable flow on a path in fixed-parameter running time. In *ICALP 2017*, pages 67:1–67:13, 2017.
- 63 Alexander Zelikovsky. An $11/6$ -approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.